움직임과 DCT 계수를 이용한 압축영역에서 MPEG 비디오의 인덱싱과 검색

박한엽[†] · 최연성[††] · 김무영[†††] · 강진석[†††] · 장경훈[†††] · 송왕철[†††] · 김장형[†††]

요 약

본 대부분의 비디오 인덱싱 어플리케이션들은 빠르고 효과적인 저장, 브라우징, 검색 기법들을 요구한다. 현재까지 연구된 많은 기법들은 그 처리 영역이 화소 영역에 대해서만 다루어왔다. 이런 접근은 멀티미디어 데이터가 일반적으로 압축된 형태로 저장되기 때문에 해독, 재압축에 따른 높은 오버헤드를 초래한다. 반면에 압축된 비디오 데이터가 주어질 때, 압축된 데이터를 직접 분석할 수 있다면 화소단위 처리에서 생기는 오버헤드를 피할 수 있다. 본 논문에서, 압축된 비디오 스트림 정보를 직접 분석하여 비디오 인덱싱에 유용한 특징들을 추출한다. 이러한 특징들을 이용하여 컷 검출을 위한 기법을 유도하여 스트림을 샷으로 분할한다. 또한 본 논문에서, 공간적 정보(DCT 계수)와 시간적 정보(움직임 벡터)를 이용하여 간단한 키 프레임 선택 기법과 효과적인 비디오 인덱싱기법을 제안한다.

Video Indexing and Retrieval of MPEG Video using Motion and DCT Coefficients in Compressed Domain

Han-yeop Park[†], Yeon-sung Choi[††], Mu-young Kim[†††], Jin-suk Kang[†††], Kyung-hoon Chang[†††], Wang-chul Song[†††] and Jang-hyung Kim[†††]

ABSTRACT

Most of video indexing applications depend on fast and efficient archiving, browsing, retrieval techniques. A number of techniques have been approached about only pixel domain analysis until now. Those approaches brought about the costly overhead of decompressing because the most of multimedia data is typically stored in compressed format. But with a compressed video data, if we can analyze the compressed data directly, then we avoid the costly overhead such as in pixel domain. In this paper, we analyze the information of compressed video stream directly, and then extract the available features for video indexing. We have derived the technique for cut detection using these features, and the stream is divided into shots. Also we propose a new brief key frame selection technique and an efficient video indexing method using the spatial informations(DCT coefficients) and also the temporal informations(motion vectors).

1. 서 론

인터넷을 비롯한 정보통신망이 급속히 대중화되며, 프로세서와 대용량 저장장치의 가격대성능비가 낮아짐으로 인해 멀티미디어가 각 분야에서 광범위하게 이용되고 있다. 정보의 전송과 저장에는 반드시 효과적인 검색이 수반되어야 하며, 이런 제반 과정을 표준화하기 위하여 MPEG-7이 제안되어 검토되고 있다.

데이터 검색을 위하여 키워드를 선정하듯이 디지털 비디오 검색에도 키 프레임이 선정되거나 물체가 추출되어 이를 바탕으로 인덱싱이 이루어져야 한다. 그러나 디지털 비디오는 대용량이기 때문에 압축을 피할 수 없고, 압축상태에서는 장면의 특징, 등장 요

본 논문의 연구는 1998 정보통신부 우수대학원 지원에 의해 수행되었음.
† 군산대학교
†† 정회원, 군산대학교
††† 제주대학교

소 등이 전혀 보존되지 않아서 인덱싱을 위한 키를 선정할 수 없다. 그러므로 압축 비디오는 해독되어 원래의 상태로 환원되어 인덱싱된 후 재차 압축되어야 한다. 압축을 위한 인코딩은 많은 연산과 시간을 요하므로 이의 반복은 결코 바람직하지 못하다. 실시간 인코딩을 위해서는 고가의 전용 프로세서가 요구되기도 한다.

비디오 인덱싱과 검색 문제는 많은 방법으로 연구가들에 의해서 언급되어 왔다. 주로 픽셀영역에서 디지털 비디오의 파싱과 인덱싱 기법들의 연구는 비디오 인덱싱의 연구경향들과 앞으로의 데이터 전달 시스템들의 요구사항들을 언급한 Ahanger와 Little[1]의 논문에서 제안되었다. 이와는 달리, Zhang[2]은 장면내용 표현의 논의에 대해서 픽셀영역에서 이용을 위한 기법을 기술하였을 뿐만 아니라, 장면들의 시간적 특성들과 키 프레임들을 이용한 내용기반 검색 기법도 논의하였다. 또한, 키 프레임 추출에 의해 얻어진 픽셀영역에서의 비디오 파싱에 대한 기법을 제안하였다. 장면내용의 표현은 컬러 히스토그램과 움직임, 텍스쳐, 형태, 에지 특징들을 포함하는 몇 가지 형태의 특징들에 기반을 둔다. 이와 비슷한 연구로는 국부적으로 유도된 특징들을 이용하여 지정된 물체들에 대한 완전한-비디오 검색을 수행하기 위한 픽셀-영역 기법들을 제안한 Nagasaka와 Tanaka[3]의 연구이다. Flicker[4]는 대량의 이미지/비디오 데이터베이스들에서 컬러, 형태, 텍스쳐, 스켓치에 기반한 내용기반 검색을 수행하는 QBIC 시스템을 기술하였다. 전술한 모든 논문들은 픽셀영역에서의 기법들을 제안하였지만, 압축된 영역에서의 비디오 인덱싱을 다룬 것은 드물다.

Idris와 Panchanathan[5]은 압축된 형태에서 비디오 시퀀스의 인덱싱을 위하여 벡터 양자화(vector quantization:VQ)에 기반한 알고리즘을 제안하였다. 압축되는 동안, 이미지는 벡터들로 분해되고 일정집합의 코드워드(codeword)들로 나타내어지고 적용된 VQ를 이용하여 인코드된다. 각 프레임은 인덱스들을 생성하기 위해 이용된 코드북(codebook)과 레이블집합에 의해 표현된다. Chang[6]에 의한 압축된 영역의 비디오 인덱싱 기법들의 총체적인 논문은 이러한 DCT, wavelet, subband transform compress domain을 어드레싱에 관계된 문제점들을 기술하였다.

본 논문과 비슷한 개념으로 [7]에서 샷 변화를 검출하기 위해 I 프레임들의 DCT 정보만을 이용하였다. 이런 기법들은 단지 I 프레임만을 갖고있는 비디오 클립(Motion JPEG)에 적합하고 비디오 클립이 I 프레임들 사이에 많은 수의 P와 B 프레임들을 가질 경우에는 부적합하다. 또한 시간적 해상도에서 축소 때문에, 샷에서 움직임을 검출하는데 실패하게 될 가능성이 크다. 만약 두 개의 연속되는 I 프레임들 사이에 하나이상의 샷 변화가 있다면, 이런 기법들은 어떤 샷 변화도 검출할 수도 없으며, 어디에서 변이가 발생하였는지도 지적할 수가 없다.

비디오 재압축에 대한 오버헤드를 없애기 위하여 본 논문에서는 변환영역, 즉, 압축된 상태에서 특징을 추출하여 인덱싱할 수 있는 새로운 기법을 제안한다. MPEG 비디오를 대상으로 했으며, 변환영역에서 DCT 계수와 움직임 벡터로부터 장면전환을 추정한다. 먼저, 2장에서 변환영역에서의 비디오 분할(segmentation) 기법에 대해 기술한다. 3장에서는 압축된 영역에서의 특징추출기법[8]과 이를 이용한 비디오 인덱싱 기법을 기술한다. 4장에서 앞에서 기술한 여러 가지의 기법들을 실험을 통해 확인하고 이를 검토한다. 마지막으로 5장에서 본 논문의 내용을 요약하고 결론을 맺는다.

2. MPEG 비디오에서의 비디오 분할

비디오 분할[9]은 한 샷에서 또 다른 샷으로 변이가 일어나는 프레임들을 검출하는 것으로 정의된다. 두 프레임들 사이에 이런 변화가 발생하는 경우 그것을 컷(cut) 혹은 브레이크(Break)라 한다. 몇 개의 프레임들에서 천천히 발생하는 변이에는 페이드(fade), 디졸브(dissolve), 와이프(wipe)가 있다. 이런 샷들은 카메라 움직임에 따라 더 세분화되어진다. 본 논문에서는 페이드, 디졸브, 와이프에 대한 검출법은 언급하지 않고 컷 검출기법만을 설명한다. 컷은 장면 전이의 일반적인 형태이며, 여타의 방법은 이의 응용이기 때문이다[7][10]. 본 기법을 이용하여 다른 카메라 동작을 검출할 수 있을 것이다[11].

2.1 매크로블록 기반 컷 검출 알고리듬

매크로블록(macroblock:MB) 정보를 이용하는 분할은 비디오 클립에 존재하는 프레임간 일관성에 근거한다. MPEG 스트림은 I, P, B 프레임들을 갖는다.

만약 한 P 프레임의 대부분이 인트라 부호화된 MB들이라면, 이것은 이런 MB들이 이전 참조 프레임(reference frame)으로부터 예측될 수 없고, 현재 P 프레임과 이전 I나 P 프레임사이의 어떤 곳에서 샷변화가 발생하였을 가능성이 매우 크다는 것을 암시한다. 만약 B 프레임에서, 대다수의 MB들이 이전 I나 P 프레임들로부터 순방향으로 예측되었다면, 이는 현재 프레임과 다음 I나 P 프레임사이에 샷변화가 발생할 가능성이 매우 높다. 또한, 만약 B 프레임에서 대다수의 MB들이 역방향으로 예측되었다면, 이전 I나 P 프레임과 현재 프레임사이에 샷변화가 발생했을 가능성이 매우 높다. 어떤 두 참조 프레임들 사이에서, 샷변화의 양쪽 B 프레임들은 위의 두 가지 경우에 따라야만 한다. 즉, 샷변화에 앞선 모든 B 프레임들에 대해서는 첫 번째 경우가 참이어야 하고, 샷변화 후의 모든 B 프레임들에 대해서는 두 번째 경우가 참이 되어야 한다. 이런 명제들이 검출과정에서 적용되었다.

본 논문에서는 각 프레임에서 각각의 MB의 타입의 수를 카운트한다. 그 참조 프레임에 유사한 정도를 나타내는 유사값(SV : Similarity Value)은 이런 카운트들로부터 얻어진다. 이 값은 시퀀스에서 브레이크를 검출하기 위해 직접적으로 이용될 수 있는 수단이다. 프레임들에 대한 SV들이 한번 결정되어지면, 각 쌍의 참조 프레임들이 순차적으로 고려되고, 이 두 참조 프레임들 사이에 발생한 어떤 브레이크들을 정확하게 나타내기 위해 이용되었다.

매크로블록 기반 검출 알고리듬은 각 프레임의 MB들의 형태를 이용하여 컷 검출 여부를 결정한다. MB들의 총 개수와 스킵된 MB들의 개수를 식 (1)과 같이 정의되고, 표 1은 이 장에서 이용된 변수들을 정의한 것이다.

$$
\begin{aligned}
N &= N_f + N_b + N_{fb} + N_s \\
N_s &= N_{sf} + N_{sb} + N_{sfb}
\end{aligned}
\qquad (1)
$$

클립에서의 I 프레임들의 집합을 $\mathbb{I}$, P 프레임들의 집합을 $\mathbb{P}$, B 프레임들의 집합을 $\mathbb{B}$로 놓자. SV_i는 수식 (2)와 같이 비디오 클립에서 각 프레임 i에 지정되고 I 프레임에서는 이전 프레임의 SV_i 값을 이용하고 P프레임에서는 순방향과 스킵된 MB들의 수를 더한 값이고 B 프레임에서는 쌍방향과 순방향으로 예측되거나 스킵된 MB들의 수로 표현하였다.

표 1. MB 변수 정의

변수	정의
N	한 프레임에서의 총 MB들의 수
N_f	순방향으로 예측된 MB들의 수
N_b	역방향으로 예측된 MB들의 수
N_{fb}	쌍방향으로 예측된 MB들의 수
N_s	스킵된 MB들의 수
N_{sf}	순방향으로 예측되어 스킵된 MB들의 수
N_{sb}	역방향으로 예측되어 스킵된 MB들의 수
N_{sfb}	쌍방향으로 예측되어 스킵된 MB들의 수

비디오 클립이 I와 P 프레임만을 갖는다면, 프레임 i에서 시작하는 새로운 시퀀스가 결정되어진다 ($i \in \mathbb{P}$, if $SV_i / N < t1$).

$$
SV_i = \begin{cases} 0 & \text{if } i = 0 \\ SV_{i-1} & \text{if } i \in \mathbb{I} \\ N_f + N_s & \text{if } i \in \mathbb{P} \\ N_{fb} + N_{sfb} + N_f + N_{sf} & \text{if } i \in \mathbb{B} \end{cases} \qquad (2)
$$

연속되는 I 프레임들의 DCT 분석은 I 프레임 전에 발생하는 샷 변화를 검출하기 위해 수행된다. DCT 분석에 의해 연속되는 I 프레임들 사이에서 컷이 검출되고 I 프레임들 사이의 P 프레임의 MB들의 형태 대부분이 순방향 예측되었다면, 후자의 I 프레임에서 시작하는 새로운 시퀀스가 결정되어진다.

한 클립에서 P 프레임들이 존재한다면, 일반적으로 B 프레임들 또한 존재한다. 비디오 클립이 B 프레임들을 갖는다면, R을 참조 프레임들의 집합이라고 할 때, $\mathbb{R} = \mathbb{I} \cup \mathbb{P}$이다. 어떤 두 참조 프레임들 R_i와 R_j 사이에 후자의 참조 프레임 R_j와 함께 모든 B 프레임들을 포함하는 집합을 도메인 $\mathbb{D}$로 정의하자.

본 절 앞부분에 언급되었던 내용을 표로 정리하면 다음과 같다. 즉, 어떤 프레임 $k \in \mathbb{D}$에 대해, 만약 표 2에 서술된 네 가지 조건들을 만족한다면, 프레임 k에서 시작하는 새로운 시퀀스가 결정되어진다. 여기서 이용된 임계값 t_1, t_2, t_3는 MB들을 이용한 컷 검출 과정에 이용된다. 그리고 t_4, t_5는 DCT 계수 비교법(DCT Coefficients Comparison)에 적용된다. 먼저, t_1은 I와 P 프레임만을 포함하는 MPEG 스트림에 이용된 임계값이다. t_2는 MPEG 프레임의 컷의 존재 검사여부를 확인하기 위해 이용되며 그 적당한 값은 0.5이고, 이 과정이 정당하면 임계값 t_3는 고정적으로 0.33을 이용한다. 임계값 t_4는 연속적인 I 프레

표 2. MB 기반의 컷 검출 조건

	조 건
조건 1	만약 $R_j \in$ P이면, $SV_j/N < t_2$
조건 2	$\forall l \in$ Đ일 때, $SV_l/N < t_3$ ($l \neq$ B)
조건 3	∀Đ에서 k이전의 B frames, $N_f - N_b > N_{fb}$
조건 4	∀Đ에서 k이후의 B frames, $N_b - N_f > N_{fb}$

임들에서 MB들 사이의 불균형을 조절하며 그 적당한 값은 200이다. 이는 매크로블록 기반의 샷 검출 알고리듬이 불충분할 때 DCT 계수 비교법에 이용된다. 임계값 t_5는 0.25를 이용한다.

만약 조건 1은 해당 프레임의 검사여부 결정에 이용된다. 만약 이 조건이 거짓이면, 남은 조건들에 대한 테스트는 모두 생략된다.

2.2 DCT 계수비교법(DCT Coefficients Comparison)

DCT 계수 비교법은 매크로블록 정보가 불충분한 경우에 대해 어떤 샷 변화의 존재 여부를 확인하기 위해 이용된다. 어떤 경우에는, 단순한 매크로블록 형태들의 분석이 두 프레임들 사이에서 샷 변화가 발생하였다는 것을 명시하기 위한 충분한 정보를 제공하지 않는다. 이런 경우들은 일반적으로 샷 변화 주위에 많은 움직임이 없고, 대부분의 MB들이 스킵되어질 때 발생한다. MPEG 스트림으로 인코드된 DCT 데이터를 분석하는 것은 비디오 클립에서 컷의 존재 여부를 확인하는데 도움을 주게 된다.

본 논문에서는 이런 난해한 경우를 정확하게 지적하고 샷 변화의 존재여부를 확인하기 위해 B 프레임들에 걸친 두 개의 I 프레임들의 DCT 계수들을 이용한다. 이 분석은 임계값 t_4보다 큰 값을 취하는 DC 계수들의 수를 카운트하는 것을 의미한다. 이 카운트가 다른 임계값 t_5를 초과하면, 이 두 프레임들 사이에서 샷 변화가 발생하였을 가능성이 크다[12]. Motion-JPEG의 경우에서 이 방법은 컷 검출을 위해 상당히 확실한 방법으로 이용될 수 있다[13].

DCT 계수 비교법은 어떤 도메인 Đ에 대하여 그 도메인에서 위의 절차에 의해 샷 변화가 검출되었고 그 도메인의 끝에 있는 참조 프레임이 I 프레임이고 Đ에서의 컷 주변 B 프레임에 대해 $N_s/N > t_4$이면, 그 도메인에 대한 DCT 분석은 다음과 같다.

$$\delta_{ij} = \begin{cases} 1 \text{ if } |DCT(I_m)^0_{ij} - DCT(I_n)^0_{ij}| > t_4 \\ 0 \text{ otherwise} \end{cases} \quad (4)$$

$$\{\sum_{ij} \delta_{ij}\}/N > t_5 \quad (5)$$

여기서 $DCT()$는 주어진 프레임의 DCT를 취하는 함수이다. $DCT(I_m)^0_{ij}$은 I_m프레임의 ij^{th} DC 계수를 나타낸 것이고 각 MB에 존재하는 6개의 DC 계수들을 더한 값이다. 만약 식 (5)가 참이면 현재 프레임의 MB들이 그 참조 프레임과 적당한 차이(t_5)를 보이는 것이기 때문에 그 프레임에서 샷 변화의 존재가 입증된다. 본 논문에서 제안한 DCT 비교법은 만약 샷 변화가 발생한 것이 아니라면, 많은 MB들이 스킵되었다는 사실을 고려한 매우 작은 움직임과 MB들의 DC 계수들의 값들이 충분히 큰 차이를 보이는 MB들의 수를 이용하여 샷 변화 여부를 결정한다. 이 DCT 계수 비교법을 이용하여 매크로블록 기반의 컷 검출의 문제점을 바로잡을 수 있다. 여기에서 t_1, t_2, t_3는 프레임간의 유사도를 나타내는 SV를 고려하여 설정한 것이며, t_4는 DC 계수들의 차분 허용치이며 t_5는 수식 (5)에서 샷 변화의 임계값을 25%로 설정하려는 의도이다.

3. 특징추출과 비디오 인덱싱

3.1 DCT 계수의 추정

DCT 계수는 I 프레임에서는 쉽게 접근할 수 있지만, P와 B 프레임이 예측이나 보간 후 오차 에러에 의해 표현되기 때문에, P와 B프레임의 DCT 계수는 추정되어야 한다. 그림 1은 MB_{Ref}에 의해 정의된 16×16 영역으로부터 예측되어진 P 프레임에서 MB_{Cur}를 보여주고 있다.

P 프레임을 인코딩하는 동안, MB_{Ref}를 고려한 MB_{Cur}의 오차 에러만이 저장된다. MB_{Ref}의 DC 계수

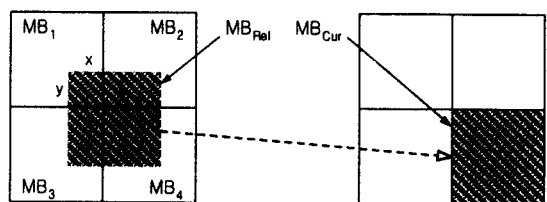



그림 1. MB_{Ref} 에서 P 프레임의 MB_{Cur} 예측

는 MB_1, MB_2, MB_3, MB_4의 DC 계수들로부터 계산되어진다. 높은 비용의 계산을 피하기 위해, 단지 DC 계수만이 네 개의 MB들의 DC 계수들의 가중된 합에 의해 추측되어진다. 여기서 가중치는 참조 MB를 덮고있는 이 MB들의 영역의 한 부분이 된다. 즉,

$$DC(MB_{Ref}) = \sum_{i=1}^{4} w_i \times DC(MB_i) \qquad (6)$$

여기서 w_i 는 그 모든 영역에 MB_i의 빗금 쳐진 지역의 영역의 비율로 주어진다. 비슷하게, B프레임에서의 MB가 두 참조 MB들에서 중간치로 결정되어진다면, 그 DC 계수는 이 두 MB들의 추정된 DC 계수들의 평균치로 추정되어진다.

3.2 움직임정보의 추정

MB는 MB가 인트라 부호화된 것인지, 순방향/역방향으로 예측된 것인지, 혹은 쌍방향으로 예측된 것인지에 상관없이 각각의 프레임 형태에 따르는 0, 1, 2개의 움직임벡터들을 가질 수 있다. 움직임벡터가 없는 I 프레임과 쌍방향 예측된 B 프레임과 비교할 때, 그리고 예측방향이 다른 두 개의 B 프레임을 비교할 때 문제가 발생한다. 여기서 예측방향과 프레임 형태에 독립적인 움직임벡터가 요구되며, 이에 본 논문에서는 각각의 움직임벡터들을 프레임 형태에 독립적으로 다음 프레임에 대해 역방향으로 예측된 벡터로 표현한다.

3.2.1 I와 P 프레임만을 갖는 MPEG 비디오

만약 I 프레임들 사이에 P 프레임들만 있다면, 흐름(flow)은 두 개의 연속되는 I 프레임들 사이의 각 프레임들에 대해서 추정될 수 있다. P 프레임에서의 MB가 I 프레임의 MB에 대한 움직임벡터 (x,y)에 의해 이동되었다면, I 프레임의 MB가 P 프레임에서의 MB에 대한 움직임벡터 (-x,-y)에 의해 이동되었다고 추정할 수 있다. 같은 방법으로 또 다른 P 프레임들이 뒤따르는 P 프레임의 MB들의 흐름추정(flow estimation)에 적용되어진다.

3.2.2 B 프레임을 포함한 MPEG 비디오

대부분의 MPEG 스트림은 연속되는 참조 프레임들 사이에 B 프레임을 갖는다. 두 개의 연속되는 참조 프레임을 R_i와 R_j 라하고, 이 참조 프레임들 사이에 B프레임들을 $B_1 ..., B_n$ 으로 나타낸다. 각 B 프레임에서 각각의 MB는 세 가지 형태 중에 하나를 취할 수 있기 때문에, 두 개의 이웃하는 B 프레임들 사이에는 9개의 가능한 MB 조합이 존재할 수 있다(FF, FB, FD, BF, BB, DF, DB, DD : F[순방향], B[역방향], D[쌍방향]). 흐름은 이 조합들 사이에 참조 프레임에 대한 각각의 움직임벡터를 분석함으로써 추정된다. 이는 움직임의 방향성을 고려하면 크게 4가지의 경우로 분류된다.

3.2.2.1 FF, FD, DF, DD의 경우

이 조합은 순방향으로 예측된 2개의 움직임 벡터만이 요구되며, B_1에서의 현재 MB의 순방향-예측된 움직임벡터를 $\overrightarrow{B_1R_i}$로 나타내고, B_2에서 부합되는 MB의 순방향-예측된 움직임벡터를 $\overrightarrow{B_2R_i}$로 나타낸다. 만약 B 프레임들의 MB들 사이에 흐름을 $\overrightarrow{B_1B_2}$로 정의하면, 다음과 같은 관계식을 갖는다.

$$\overrightarrow{B_2R_i} = -\overrightarrow{B_1R_i} + \overrightarrow{B_1B_2} \qquad (7)$$

3.2.2.2 BB, BD, DB의 경우

이 조합은 위의 조합과는 정반대의 경우로 각 MB의 역방향 움직임 벡터만을 요구하는 경우이다. R_j에 대한 B_1과 B_2의 역방향-예측된 움직임벡터를 이용하여 $\overrightarrow{B_1R_2}$를 얻을 수 있다. B_1에서의 현재 MB의 역방향-예측된 움직임벡터를 $\overrightarrow{B_1R_i}$로 나타내고, B_2에서 부합되는 MB의 역방향-예측된 움직임벡터를 $\overrightarrow{B_2R_j}$로 나타낸다. 그러면 다음과 같은 관계식을 얻을 수 있다.

$$\overrightarrow{B_1R_j} = \overrightarrow{B_1B_2} + \overrightarrow{B_2R_j} \qquad (8)$$

3.2.2.3 BF의 경우

BF의 경우에 대해서는 두 MB의 참조 MB이 같지 않기 때문에 수식 (8)을 이용할 수 없다. MB가 순방향으로 예측된 확실한 움직임벡터를 갖는 프레임(B_k)을 찾을 때까지의 연속적인 B 프레임들을 살펴보고, 그 벡터의 역을 이용한다. 이 벡터가 이전에 k개의 프레임들로부터 예측되어졌기 때문에, 인수 k를 이용하여 그것을 감소시키며 스케일(scale-down)한다. 만약 그 같은 B 프레임을 찾을 수 없었다면, 정의되지 않은 것으로 그 흐름벡터(flow vector)를 스

킵한다. 이러한 방법으로 R_2에 일치하는 MB에 대한 흐름벡터 $\overrightarrow{R_iB_1}'$를 알아낸 다음에 B_2의 순방향-예측된 움직임벡터 $\overrightarrow{B_2R_i}$를 이용하여 $\overrightarrow{B_1B_2}$를 다음과 같이 계산한다.

$$-\overrightarrow{B_2R_i} = \overrightarrow{R_iB_1}' + \overrightarrow{B_1B_2} \tag{9}$$

비슷하게, 스케일 감소 기법을 이용하여 R_j에 일치하는 B_2에서의 MB에 대한 흐름벡터 $\overrightarrow{B_2R_j}'$를 알아낸다. 그러면, B_1의 역방향-예측된 움직임벡터 $\overrightarrow{B_1R_j}$를 이용하여 $\overrightarrow{B_1B_2}$를 다음과 같이 계산한다.

$$\overrightarrow{B_1R_j} = \overrightarrow{B_1B_2} + \overrightarrow{B_2R_j}' \tag{10}$$

벡터들이 스케일 감소 기법을 이용하여 참조 프레임들에 대해서 추정되어왔기 때문에, 실제적인 $\overrightarrow{B_1B_2}$의 더욱 적합한 추정값을 얻기 위해 이들 두 벡터들의 평균을 취한다.

3.2.2.4 FB의 경우

두 B 프레임들 사이에 컷이 존재해 정의가 불가능하다.

3.3 DCT 계수와 움직임에 의한 인덱싱

MPEG으로 인코드된 일련의 비디오 정보가 주어진다면, 다양한 질의들이 가능하도록 인덱스하는 것이 좋다[14]. 본 논문에서, 각 장면들의 키 프레임은 공간적 차원을 표현하는 DC 계수들과 시간적 차원을 표현하는 흐름벡터들을 그 장면에 대한 인덱스로 이용하였다. 그림 2는 MPEG 비디오에서 컷 검출에 의해 분해된 샷을 인덱싱하는 과정을 간략하게 보여준다.

3.3.1 키 프레임 선택

2장에 기술한 컷 검출 알고리듬을 이용하여 일정한 움직임을 갖는 한 샷으로 그룹화 된다. 이러한 샷들 중에서 잡음 때문에 분할된 것이 아닌 짧게 끝나는 유사 움직임 시퀀스(similar motion sequence)들은 더 긴 시퀀스들을 만들기 위해 같이 그룹화 된다.

키 프레임들은 각각의 이런 서브샷들에 대한 대표 프레임(representative)들로서 선택되어진다. 키 프레임들은 저장과 검색 체제의 중요한 블록들이 되기

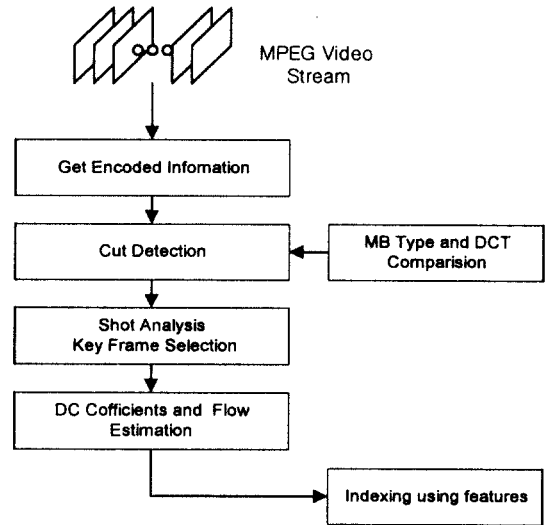



그림 2. MPEG에서의 비디오 인덱싱

때문에, 키 프레임에 대한 최적의 대표 프레임은 I 프레임이다[13]. 이렇게, 새로운 샷의 첫 번째 I 프레임은 그 샷의 키 프레임으로써 선택된다. 그런데, 첫 번째 I 프레임이 그 샷에 너무 멀리 떨어진 경우가 있다. 이런 경우에 P 프레임이 선택될 수 있고, P 프레임은 DCT 계수들을 포함하고 있지 않기 때문에 전에 언급했던 기법을 이용하여 추정되어진다.

3.3.2 공간적 유사성

두 프레임들 사이의 공간적 유사성은 프레임들이 휘도, 색도, 질감, 형상과 같은 비슷한 공간적 특성들을 갖는다는 것을 의미한다. 공간적 유사성을 찾는데 가장 간단한 접근방법은 모든 시퀀스의 매 프레임에 대한 모든 DCT 정보를 저장하고 비교 검색하는 것이다. 질의 프레임이 주어질 때, 가장 비슷한 한 프레임을 찾기 위해 가능한 한 모든 정보와 비교해야할 것이다. 그러나, 이 접근방법은 시간과 공간 모두에서 비효율적이다. 본 연구에서는 보다 효과적인 검색을 위해 비디오 데이터들의 속성들을 이용한다.

전에 언급된 것처럼, 각 비디오 스트림은 샷들로 나누어질 수 있다. 그리고 각 샷은 비슷한 프레임들의 가장 기본적인 시퀀스임을 의미한다. 각 샷에 대하여 키 프레임이 선택되어지고, 그 키 프레임은 질의가 그 샷을 검색할 것인가 아닌가를 결정하는데 이용된다. 문제는 개개의 프레임들 사이에 유사성을 어떻게 결정하느냐는 것이다. 본 논문에서의 접근방

법은 직접 해당 프레임들의 부합하는 DCT 계수를 비교하는 것이다. 왜냐하면 압축된 영역에서는 DCT 계수들이 개개의 프레임들의 공간적 정보를 가장 잘 표현하고 있기 때문이다. 그리고 DC 계수는 그 블록에 대한 평균적인 명암을 나타낸다. 본 논문에서는 각 프레임들을 압축, 해제하는 것을 피하기 위해, 휘도와 색도성분의 DC 계수만을 이용하고 공간적 유사성을 분석하기 위해 그 계수들과 다른 프레임에서 부합하는 DC 계수들을 비교한다. 인덱싱과정을 정리하면, 352×288의 MPEG 비디오에 대해 먼저 스트림을 샷들로 분해하고 키 프레임을 결정하여 그 키 프레임에서 6개의 DC 계수들과 각각의 움직임벡터를 이용하고, 흐름정보를 위해 뒤따르는 프레임들 사이의 단순한 거리함수를 취한 값으로 인덱싱을 수행한다. 적응된 B^+ 트리[15,16]를 이용하여, 각각의 샷들에 대해 키 프레임의 MB들의 DC 계수들과 움직임벡터를 인덱싱한다. 보다 빠른 검색을 위해 키 프레임을 비교하기 전에 각 노드들은 키 프레임의 4개의 휘도성분의 DC 계수들의 평균적인 합과 2개의 색도성분의 DC 계수들의 평균적인 합을 저장하여 검색시에 이를 비교하여 타당성을 검사하는데 이용하고, 실질적인 DC 계수 비교는 컷 검출에 이용하였던 식 (4)와 식 (5)를 이용한다. 만약 식 (5)가 참이면 부합하지 않는 것으로, 거짓이면 부합하는 프레임으로 간주하여 나열되며 이중에서 최적으로 부합하는 것을 선택한다.

3.3.3 시간적 유사성

많은 비디오 클립들은 상당히 비슷한 내용의 샷을 갖는다. 만약 비디오 클립형태로 질의가 일어난다면 질의 키 프레임(query key frame)에 일치하는 매크로블록들의 움직임정보를 비교하여 인덱싱할 수 있다. 그리고 가장 비슷한 움직임을 갖는 키 프레임을 찾기 위해 그리고 최적으로 일치하는 키 프레임을 넘겨주기 위해 공간적 정보를 비교한다. 본 논문의 초점은 잡음과 작은 변화들에 덜 민감한 움직임의 방향에 두고 있다.

전에 언급했던 것처럼, MPEG 스트림들은 인코딩의 한 부분으로 움직임을 제공한다. 많은 클립들은 상당히 비슷한 내용의 샷을 갖는다(예를 들어, 대화장면들). 한 클립에서 키 프레임들은 각기 다른 지점에서 같은 내용이나 액션을 갖는 프레임들을 대표하기 위해 만들어 진다. 만약 비디오 클립의 형태에서 "이 짧은 비디오 장면을 포함하는 비디오 클립들을 검색하라"와 같은 질의가 발생한다면, 질의 키 프레임의 매크로블록에 부합하는 비디오클립의 움직임 정보를 비교할 수 있다. 그리고 가장 비슷한 움직임을 갖는 키 프레임을 찾기 위해 일련의 키 프레임들이 이전 장에서 언급했던 기법에 의해서 간략하게 나열되고 부합률 75%를 넘는 키 프레임 중에서 최적으로 매칭되는 키 프레임을 돌려준다.

본 논문에서 각 샷들을 대표하는 키 프레임들은 I 프레임이다(어떤 경우에서는 P 프레임). 그리고 이 전제를 타당하게 되도록 클립들이 참조 프레임들 사이에 B 프레임들을 갖는다는 것을 가정한다. 이렇게 각 키 프레임은 그것을 뒤따르는 B 프레임을 갖고 그것의 MB들은 키 프레임에 대한 순방향 움직임벡터들이다. 이런 순방향 움직임벡터들의 역을 취한 것들은 키 프레임들의 각 MB들의 역방향 움직임벡터로 간주한다. 이런 흐름정보는 그 키 프레임의 움직임을 추정하고 흐름 비교(flow comparison)에 이용된다. 키 프레임에 뒤따른 B 프레임의 모든 MB들이 순방향 움직임벡터들이 아니기 때문에, 그 흐름이 정의되지 않는 MB들이 존재하게 된다. 그러나 그러한 MB들의 수는 흐름 비교를 방해할 만큼 충분히 크지 않으므로 무시될 수 있다.

이와 같이, 2장에서 기술된 기법들을 이용하여 MPEG 스트림을 샷으로 분해한 후에 각 키 프레임에서 이 두 가지의 특징을 인덱스한다.

4. 실험결과

MPEG 스트림에서 간단하고 효과적인 인덱싱 방법을 제공하기 위해 본 논문에서는 공간과 시간적 유사성 기법을 결합하였다. 먼저, 각 MB마다 6개의 DC 계수들로부터 휘도성분의 평균치와 색도성분의 평균치, 그리고 각 DC 계수들을 적응된 B^+트리에서 키 프레임으로 인덱스한다. 이런 DC 계수들은 질의 프레임이 비디오 스트림의 I 프레임이면 손쉽게 이용가능하다. 만약 P 프레임이면 계수들은 전에 언급했던 기법을 이용하여 추정되어진다. 질의가 단 하나의 프레임으로 구성되어진다면 질의에 가장 비슷한 키 프레임을 찾기 위해 DB에 저장된 휘도성분과 색도성분의 평균치를 비교하고, 그 비교치가 어느 임계값

에 이르면 정확한 매칭을 위해 각 DC 계수들을 비교한다. 이 때는 움직임벡터를 추정할 필요는 없다.

실험을 위해, 본 논문에서는 5~30 fps(frame per second)으로 변화하는 프레임율로 디지털화한 대략 748 프레임들을 갖는 총 16 개의 비디오를 이용하였다. 본 실험에서는 Visual C++ 환경에서 수행되었다.

4.1 MPEG에서 컷 검출 실험

그림 3은 연속되는 네 개의 프레임에서 컷이 발생한 예를 보여주고 있는데, 여기서는 조건 3과 조건 4에 의해 B_2 프레임에서 컷이 검출되었다. 새로운 시퀀스는 다음 P 프레임과 거의 모든 부분이 비슷한 B_2 프레임에서 시작한다. 표 3은 그림 3에서 나타난 각 프레임에 따른 조건 값들을 보여준다. 조건 1과 조건 2는 동일한 값(SV_i/N)을 이용하여 t_2와 t_3에 각각 비교한 것이고 조건 3과 조건 4는 B 프레임들을 대상으로 한 것이다.

4.2 DC 계수와 움직임정보의 추정실험

MPEG 시퀀스가 컷 검출에 의해 여러 개의 샷으로 분해되어지고 그 샷들을 대표하는 키 프레임을 선택한다. 이 절에서는 각 키 프레임들의 특징인 DC 계수들과 흐름정보를 추정한 결과를 보여준다. 그림 4는 P 프레임에서 추정된 DC 계수에 의해 복원된 영상과 압축을 해제한 후에 복원된 영상을 보여주고 있고, 그림 5는 본 DC 계수의 추정기법이 원래의 DC 계수들과 큰 차이를 보이지 않는다는 것을 보여주고 있다.

움직임정보(흐름정보)는 본 인덱싱을 위한 두 번째 특징으로 그림 6은 네 개의 프레임에서 추정된 각각의 흐름을 보여주고 있다.

4.3 MPEG 비디오 질의실험

DB에 저장된 키 프레임의 DC 계수들과 움직임

(a) B_2 Frame

(b) I Frame

(c) B_1 Frame

(d) B_2 Frame

Intra MBs　Forward MBs
Backward MBs　Bidirectionally MBs
Skipped MBs

그림 3. MPEG 비디오에서 샷변화 검출

(a) original DC　(b) estimated DC

그림 4. 추정된 DC 계수에 의해 복원된 영상비교

표 3. 조건 값 결과

	SV_i/N	SV_j/N	$N_{f(i-1)}-N_{b(i-1)}$	$N_{fb(i-1)}$	$N_{b(i)}-N_{f(i)}$	$N_{fb(i)}$
B_2	Passed	Passed	27	317	16	307
I	Passed	1.77	Passed		Passed	
B_1	Passed	Passed	Passed		-374	8
B_2	Passed	Passed	**374**	**8**	**395**	1

(a) Luminance of comparison (b) Chrominance of comparison

그림 5. 추정된 DC와 원래의 DC 값의 비교결과

(a) flow of B_2 Frame

(b) flow of P Frame

(c) flow of B_1 Frame

(d) flow of B_2 Frame

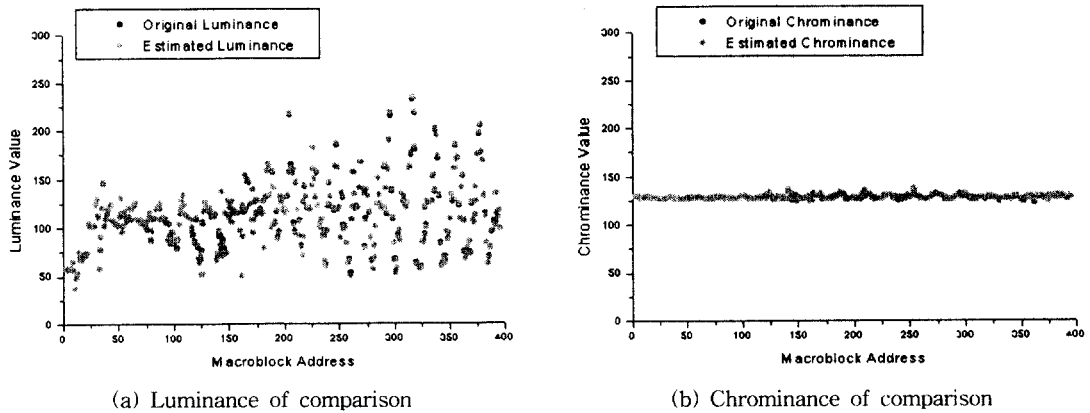
그림 6. 추정된 각 프레임의 흐름정보

정보를 비교하고 질의 키 프레임의 DC 계수들과 움직임 정보를 비교하여 비슷하게 매칭하는 시퀀스들을 돌려준다. 검색 시간을 줄이기 위해, 먼저 적응된 B^*트리의 각 노드에 저장된 DC 계수들과 움직임벡터의 평균치를 비교한 후, 비슷하게 일치하면 더욱 정확한 질의결과를 산출하기 위해 그 샷을 검색한다. 그림 7은 하나의 질의 프레임이 주어졌을 때, DB에 인덱스된 정보와 추출된 질의 프레임의 특징들을 비교하여 비슷한 프레임들을 돌려주는 것을 보여준다.

그림 8은 본 실험에서 사용한 16개의 MPEG 스트림을 이용하여 8개의 MPEG 스트림에 대한 질의 에러율과 나머지 8개의 MPEG 스트림에 대한 질의결과의 정확성을 보여주고 있다. 본 논문의 인덱싱 기법의 에러율은 10% 이내로 제한되고 성공률 또한 90% 이상임을 나타내고 있다.

그림 9는 제안한 기법들을 이용하여 화소영역과 압축된 영역에서 특징추출 후에 인덱싱하는 과정까지의 소요된 시간을 보여주고 있다. 화소영역에서 인덱싱하는 것은 MPEG 비디오를 압축 해제한 후에 그 실제적인 처리가 이루어지고 그 처리단위가 화소단위이므로 그 만큼 많은 시간이 소요되었다.

5. 결 론

본 논문에서는 압축된 MPEG 스트림의 디코딩 과정 없이 압축된 영역에서 비디오의 인덱싱과 검색기법을 제안하였다. 인덱스 되려는 MPEG 비디오는 샷들로 분해(parse)되고 그 샷들을 대표하는 각각의 키 프레임들을 선택하여 공간적인 정보와 시간적인 정보로 그 프레임을 인덱스한다.

Query Frame　　Match 1(Frame 212)

Match 2(Frame 241)　　Match 3(Frame 254)

그림 7. MPEG 비디오에서 질의 결과

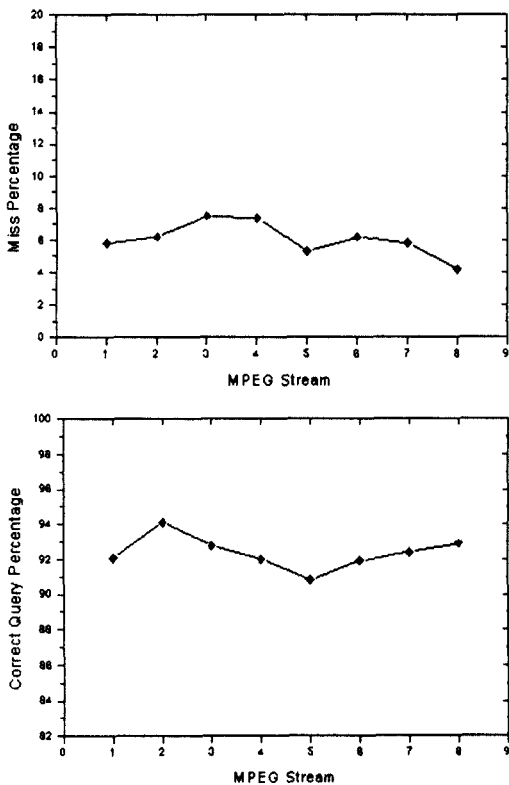



그림 8. 질의 에러율과 성공율의 비교

공간적인 정보와 시간적인 정보를 추출하기 전에 비디오 스트림은 컷 검출에 의해 샷으로 분할된다. 컷 검출은 각 프레임의 MB들의 형태에 따라 달라지며 위에서 언급한 것처럼 유사값(Similarity Value)

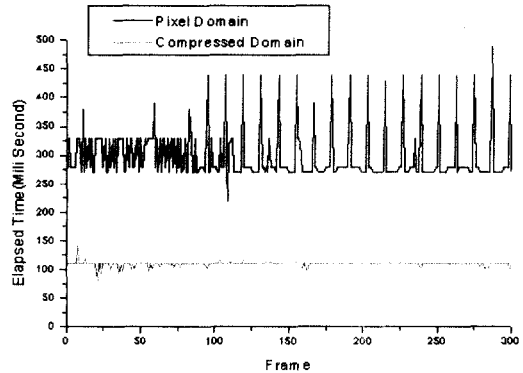



그림 9. 비디오 인덱싱 처리시간 비교

을 이용한다. MB형태가 컷 검출에 불충분하면 DCT 비교 과정을 거치게되므로 더욱 정확한 샷 검출을 가능하게 한다. 또한 본 논문에서는 I, P, B 프레임들이 동등한 것으로 간주하고, 그 때문에 압축된 MPEG 스트림의 각 프레임에 대한 여러 가지 제한들을 극복할 수 있다. 이는 지금까지 언급했던 프레임간의 DC 계수와 움직임벡터를 추정함으로써 해결되었다.

지금까지 기술되었던 컷 검출 기법과 DC 계수와 흐름정보 추출기법을 이용하여 인덱싱을 수행한 결과 위에서 보인 것처럼 90% 이상의 질의 정확성을 보였다. 이는 압축된 영역에서 압축해제없이 부호화된 정보만을 이용한 본 기법이 압축된 영역에서 매우 효과적이라는 것을 입증한다. 그리고 이러한 기법들을 이용하여 보다 대용량의 영상 데이터베이스에 적용할 수 있다.

참 고 문 헌

[1] G. Ahanger and T.D.C. Little. "A Servey of technologies for parsing and indexing digital video". *Journal of Visual Communication and Image Representation*, 7 : 28-43, 1996.

[2] H.J. Zhang, C.Y. Low, S.W. Smoliar, and J.H. Wu. "Video parsing, retrieval and browsing: An integrated and content-based solution". *In Proc. of the ACM Multimedia Conference*, pages 15-24, 1995.

[3] A. Nagasaka and Y. Tanaka. "Automatic video indexing and full-video search for object appearances". *In Proc. of the Working Con-*

ference on Visual Database Systems, pages 119-133, 1991.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. "Query by image and video content: the QBIC system". *IEEE Computer*, 28(9) : 23-32, September 1995.

[5] F. Idris and S. Panchanathan. "Indexing of compressed video sequences". *In Proc. of the SPIE on Storage and Retrieval for Still Image and Video Database IV*, volume 2670, pages 247-253, 1996.

[6] S.F. Chang. "Compress-domain techniques for image/video indexing and manipulation". *In Proc. of the IEEE International Conference on Image Processing*, volume 1, pages 314-317, 1995.

[7] F. Arman, A. Hsu, and M.Y. Chiu, "Image processing on compressed data for large video databases", *Proc. of the ACM Multimedia Conference*, pp. 267-272, 1993.

[8] K. Shen and J. Delp, "A fast algorithm for video parsing using MPEG compressed sequences", *Proc. of the IEEE International Conference on Image Processing*, Vol.2, pp. 252-255, 1995.

[9] M. Cherfaoui and C. Bertin, "Temporal segmentation of videos: A new approach", *SPIE*, Vol.2419, pp. 234-237, 1992.

[10] J. Maeda and S.F. Chang, "Tools for compressed-domain video indexing and editing", *SPIE*, Vol.2187, pp. 56-57, 1996.

[11] F. Idris and S. Panchanathan, "Indexing of compressed video sequences", *In Proc. of the SPIE on Srotage and Retrieval for Still Image and Video Databases IV*, Vol.2670, pp.247-253, 1996.

[12] E. Ardizzone, M. La Casia, V. Di Ges and C. Valenti, "Content-based indexing of image and video databases by global and shape features", *Proc. of the International Conference Pattern Recognition*, pp.140-144, 1996.

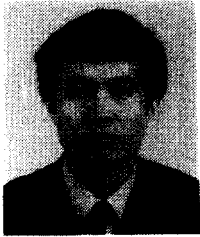
[13] B.L. Yeo and B. Liu, "Unified approach to temporal segmentation of motion JPEG and MPEG video", *International Conference on Multimedia Computing and Systems*, pp. 2-13, 1995.

[14] A. Hampapur, R. Jain and T. Weymouth, "Digital video segmentation", *Proc. of the ACM Multimedia Conference and Exposition*, pp. 357-364, 1994.

[15] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *Data Structures and Algorithms*, Addison-Wesley Publishing Company, 1987.

[16] R. Bayer and C. McCreight, "Organization and Maintenance of Large Ordered Indexes", *Acta Information 1*, pp. 173-189, 1972.

박 한 엽

1997년 군산대학교 정보통신공학과(학사)
1999년 군산대학교 대학원 정보통신공학과(석사)
2000년 군산대학교 대학원 정보통신공학과(박사과정)
1998년~현재 (주)파스칼소프트 개발2팀장
관심분야 : 영상통신, 비디오 코딩

최 연 성

1988년~1990년 제주대학교 정보공학과 조교수
1991년~현재 군산대학교 전자정보공학부 부교수
1995년 3월~1997년 2월 군산대학교 전자계산소장
1998년~현재 (주)파스칼소프트 대표이사
관심분야 : 영상처리, 멀티미디어통신
E-mail: yschoi@kunsan.ac.kr

김 무 영

1999년 2월~현재 제주대학교 정보공학과 대학원 박사과정
관심분야 : 멀티미디어

강 진 석

1999년 9월~현재 제주대학교 통신컴퓨터공학부 조교
1999년 3월~현재 제주대학교 정보공학과 석사과정
관심분야 : 멀티미디어

장 경 훈

2000년 3월~현재 제주대학교 전자공학과 박사과정
1999년 3월~현재 제주관광전문대학 전임강사
관심분야 : 전자공학

송 왕 철

1996년 2월~현재 제주대학교 통신컴퓨터공학부 조교수
관심분야 : 망관리

김 장 형

1995년 2월~현재 제주대학교 통신컴퓨터공학부 교수
2000년 3월~현재 제주대학교 통신컴퓨터공학부 컴퓨터전공 주임교수
1998년 8월~현재 제주대학교 전자계산소장
관심분야 : CAD/CAM , 멀티미디어